

AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A method comprising:
issuing an instruction selected from a queue;
enqueueing the instruction issued within a
recirculation queue;
selectively setting a state of the instruction in the
recirculation queue to one of a blocked state if completion of
the instruction is prevented by a first detected blocking
condition and an unblocked state if completion of the
instruction is prevented by a second detected blocking
condition,
wherein the first detected blocking condition
corresponds to a first data hazard, and
wherein the second detected blocking condition
corresponds to a second data hazard; and
reissuing the instruction from the recirculation queue
if a third detected blocking condition of at least one
instruction within the recirculation queue, other than the
instruction, is satisfied.

2. (Original) The method of claim 1, wherein issuing comprises:

arbitrating between a plurality of queues to select a queue;

selecting a current instruction from the queue selected; and

issuing the current instruction for the queue selected.

3. (Previously Presented) The method of claim 2, wherein issuing the current instruction comprises:

determining a state of the current instruction;

selecting an alternate queue from the plurality of queues if the state of the instruction is blocked; and

issuing an instruction selected from the alternate selected queue.

4. (Currently Amended) The method of claim 1, wherein enqueueing comprises:

detecting a blocking condition that corresponds to the first detected blocking condition prohibiting the instruction issued from completion;

placing the instruction within the recirculation queue;

setting the state of the instruction as blocked to prohibit reissue of the instruction; and
storing the first detected blocking condition.

5. (Currently Amended) The method of claim 1, further comprising:

identifying blocking conditions of instructions within the recirculation queue;

determining whether any blocking condition of any instruction within the recirculation queue is satisfied; and

enabling recirculation of instructions from the recirculation queue by setting the state of each instruction within the recirculation queue to ~~an~~ the unblocked state if any blocking condition is satisfied.

6. (Currently Amended) The method of claim 1, wherein reissuing instructions comprises:

receiving a request to issue an instruction contained within the recirculation queue;

determining a state of a current instruction of the recirculation queue;

issuing the current instruction if the state of the current instruction is ~~an~~ the unblocked state in response to the received request; and

disregarding the request if the state of the current instruction is ~~a~~the blocked state.

7. (Currently Amended) The method of claim 1, wherein enqueueing comprises:

determining whether one of the first and second detected blocking conditions preventing the instruction issued from completion is a transient blocking condition;

setting the state of the instruction to ~~an~~the unblocked state if the one of the first and second detected blocking conditions is transient; and

resetting a state of each instruction within the recirculation queue to ~~an~~the unblocked state.

8. (Previously Presented) The method of claim 1, wherein reissuing the instruction comprises:

issuing an unblocked instruction in response to a received request;

enqueueing the reissued instruction if a blocking condition of the instruction remains unsatisfied;

setting the state of the reissued instruction to the blocked state; and

storing the blocking condition.

9. (Currently Amended) The method of claim 1, wherein one of the first and second detected blocking conditions is one of a data blocking condition and a resource blocking condition.

10. (Original) The method of claim 1, wherein the recirculation queue is a first in, first out circular queue.

11. (Currently Amended) An article of manufacture including a machine readable storage medium having stored thereon instructions which may be used to program a system to perform a method, comprising:

issuing an instruction selected from a queue;

enqueueing the instruction issued within a recirculation queue;

selectively setting a state of the instruction in the recirculation queue to one of a blocked state if completion of the instruction is prevented by a first detected blocking condition and an unblocked state if completion of the instruction is prevented by a second detected blocking condition,

wherein the first detected blocking condition corresponds to a first data hazard, and

wherein the second detected blocking condition corresponds to a second data hazard; and

reissuing the instruction from the recirculation queue if a third detected blocking condition of at least one instruction within the recirculation queue, other than the instruction, is satisfied.

12. (Original) The article of manufacture of claim 11, wherein issuing comprises:

arbitrating between a plurality of queues to select a queue;

selecting a current instruction from the queue selected; and

issuing the current instruction from the queue selected.

13. (Previously Presented) The article of manufacture of claim 12, wherein issuing comprises:

determining a state of the current instruction;

selecting an alternate queue from the plurality of queues if the state of the instruction is blocked; and

issuing an instruction selected from the alternate selected queue.

14. (Currently Amended) The article of manufacture of claim 11, wherein enqueueing comprises:

detecting the first detected blocking condition prohibiting the instruction issued from completion;

placing the instruction within the recirculation queue;

setting the state of the instruction as blocked to prohibit reissue of the instruction; and

storing the first detected blocking condition.

15. (Currently Amended) The article of manufacture of claim 11, wherein the method further comprises:

identifying blocking conditions of instructions within the recirculation queue;

determining whether any blocking condition of any instruction within the recirculation queue is satisfied; and

enabling reissuing of instructions from the recirculation queue by setting a state of each instruction within the recirculation queue to ~~an~~ the unblocked state if any blocking condition is satisfied.

16. (Currently Amended) The article of manufacture of claim 11, wherein reissuing the instruction comprises:

receiving a request to issue an instruction contained within the recirculation queue;

determining a state of a current instruction of the recirculation queue;

issuing a current instruction if the state of the current instruction is ~~an~~ the unblocked state; and

disregarding the request if the state of the current instruction is the blocked state.

17. (Currently Amended) The article of manufacture of claim 11, wherein enqueueing comprises:

determining whether at least one of the first and second detected blocking conditions preventing the instruction issued from completion is a transient blocking condition;

setting the state of the instruction to the unblocked state if the at least one of the first and second detected blocking condition is transient; and

resetting a state of each instruction within the recirculation queue to the unblocked state.

18. (Previously Presented) The article of manufacture of claim 11, wherein reissuing instruction comprises:

issuing an unblocked instruction in response to a received request;

enqueueing the reissued instruction if a blocking condition of the instruction remains unsatisfied;

setting a state of the reissued instruction to the blocked state; and

storing the blocking condition.

19. (Original) The article of manufacture of claim 11, wherein the detected block condition is one of a data blocking condition and a resource blocking condition.

20. (Original) The article of manufacture of claim 12, wherein the recirculation queue is a first in, first out circular queue.

21. (Currently Amended) An apparatus, comprising:
a received instruction queue to store received instructions;
a recirculation queue;
arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction; and
blocked instruction detection logic to identify instructions blocked from execution by detected blocking conditions, and to enqueue the instructions onto the recirculation queue in one of a blocked state and an unblocked

state, including a respective blocking condition of each instruction within the recirculation queue,

wherein instructions having a transient blocking condition are enqueued onto the recirculation queue in the unblocked state, and

wherein the transient blocking condition corresponds to a data hazard.

22. (Currently Amended) The apparatus of claim 21, wherein the blocked instruction detect logic further comprises:

blocked condition satisfaction logic to:

detect whether a blocking condition of an instruction within the recirculation queue is satisfied, and to

set a state of each instruction within the recirculation queue to the unblocked state if the blocking condition of the instruction within the recirculation queue is satisfied.

23. (Currently Amended) The apparatus of claim 21, wherein the arbitration logic to:

determine a state of a selected instruction,

select the received instruction queue if the state of the selected instruction is blocked, and

issue an instruction selected from the received instruction queue.

24. (Currently Amended) The apparatus of claim 21, wherein the blocked instruction detect logic to:

determine whether the detected blocking condition is a transient blocking condition,

set a state of the instruction placed within the queue to the unblocked state if the detected blocking condition is transient, and

reset a state of each instruction within the recirculation queue to the unblocked state to enable reissue of instructions contained within the recirculation queue.

25. (Currently Amended) The apparatus of claim 21, wherein the blocked instruction detect logic to:

enqueue a reissued instruction if a blocking condition of the instruction remains unsatisfied,

~~to~~ set a state of the reissued instruction to the blocked state, and

~~to~~ store the blocking condition.

26. (Currently Amended) A system comprising:

a memory controller coupled to a memory;

a processor coupled to the memory via a bus, the processor including:

a bus interface unit coupling an execution core to a cache memory including:

a received instruction queue to store received instructions,

a recirculation queue,

arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction, and

blocked instruction detection logic to:

identify instructions blocked from execution by detected blocking conditions, and ~~to~~

enqueue the instructions onto the recirculation queue by selectively setting states of the instructions in the recirculation queue to one of a blocked state and an unblocked state, including a respective blocking condition of each instruction within the recirculation queue,

wherein instructions having a transient blocking condition are enqueued onto the recirculation queue in the unblocked state, and

wherein the transient blocking condition corresponds to a data hazard.

27. (Currently Amended) The system of claim 26, wherein the blocked instruction detect logic further comprises:

blocked condition satisfaction logic to:

detect whether a blocking condition of an instruction within the recirculation queue is satisfied, and ~~to~~

set the states of each of the instructions within the recirculation queue to the unblocked state if a blocking condition of an instruction within the recirculation queue is satisfied.

28. (Currently Amended) The system of claim 26, wherein the arbitration logic to:

determine a state of a selected instruction,

select the received instruction queues if the state of the selected instruction is blocked, and

issue an instruction selected from the received instruction queue.

29. (Currently Amended) The system of claim 26, wherein the blocked instruction detect logic to:

determine whether the detected blocking condition is a transient blocking condition,

set the states of the instructions placed within the queue to the unblocked state if the detected blocking condition is transient, and

reset the state of each instruction within the recirculation queue to the unblocked state to enable reissue of the instructions contained within the recirculation queue.

30. (Currently Amended) The system of claim 29, wherein the blocked instruction detect logic to:

enqueue the reissued instructions if a blocking condition of the instructions remains unsatisfied, ~~to~~

set the states of the reissued instructions to the blocked state, ~~and to~~

store the blocking condition.

31. (Currently Amended) A method comprising:

issuing an instruction selected from a queue;

enqueueing the instruction issued within a recirculation queue;

selectively setting a state of the instructions in the recirculation queue to one of a blocked state if completion of the instruction is prevented by a first detected blocking condition and an unblocked state if completion of the

instruction is prevented by a second detected blocking condition,

wherein the first detected blocking condition corresponds to a first data hazard, and

wherein the second detected blocking condition corresponds to a second data hazard;

resetting the state of the instruction within the recirculation queue if a third detected blocking condition of at least one instruction within the recirculation queue, other than the instruction, is satisfied; and

reissuing the instruction from the recirculation queue if the state of the instruction is indicated as the unblocked state.

32. (Currently Amended) A method comprising:

issuing a first instruction from a queue;

detecting a first blocking condition for the first instruction prior to execution of the first instruction;

setting the first instruction to one of a blocked state when the first blocking condition corresponds to a first data hazard and an unblocked state when the first blocking condition corresponds to a second data hazard~~based on the first blocking instruction;~~

enqueueing the first instruction within a recirculation queue;

selectively setting a state of the instruction in the recirculation queue to one of the blocked state and the unblocked state if completion of the instruction is prevented by the first blocking condition; and

reissuing the first instruction from the recirculation queue if the first blocking condition is satisfied.

33. (Previously Presented) The method of claim 32 further comprising:

detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction; and

reissuing the first instruction from the recirculation queue if the second blocking condition is satisfied.

34. (Previously Presented) The method of claim 32 further comprising:

setting the first instruction to the unblocked state based on the first blocking condition; and

enqueueing the first instruction within the recirculation queue in the unblocked state until the first blocking condition is satisfied.

35. (Previously Presented) The method of claim 34 further comprising:

detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction;

setting the second instruction to the blocked state based on the second blocking condition; and

enqueueing the second instruction within the recirculation queue in the blocked state until the second blocking condition is satisfied.

36. (Previously Presented) The method of claim 32 wherein enqueueing comprises:

determining whether the first blocking condition is a transient blocking condition; and

setting the first instruction to the unblocked state if the first blocking condition is transient.

37. (Previously Presented) The method of claim 1, wherein selectively setting the state of the instruction in the recirculation queue to the unblocked state is based on whether the detected blocking condition is a transient blocking condition.

38. (Previously Presented) The method of claim 37, further comprising delaying reissue of the instruction from the recirculation queue when the instruction is in the unblocked state.

39. (New) The method of claim 1,
wherein the first detected blocking condition corresponds to a non-transient blocking condition based on the first data hazard,

wherein the second detected blocking condition corresponds to a transient blocking condition based on the second data hazard, and

wherein the first data hazard is different than the second data hazard.

40. (New) The method of claim 39, wherein the second data hazard is based on a plurality of instructions that include the

instruction, wherein the plurality of instructions are issued to a single address.

41. (New) The method of claim 39, wherein the first data hazard is based on at least one of memory latency and bandwidth.